Title
Behavior-Based Artificial Intelligence

Journal
Proceedings of the Annual Meeting of the Cognitive Science Society, 15(0)

Author
Maes, Pattie

Publication Date
1993

Peer reviewed

Behavior-Based Artificial Intelligence

Pattie Maes
MIT Media-Laboratory
20 Ames Street Rm 401
Cambridge, MA 02139
pattie@media.mit.edu

Abstract

This paper attempts to define Behavior-Based Artificial Intelligence (AI) as a new approach to the study of Intelligence. It distinguishes this approach from the traditional Knowledge-Based approach in terms of the questions studied, the solutions adopted and the criteria used for success. It does not limit Behavior-Based AI to the study of robots, but rather presents it as a general approach for building autonomous systems that have to deal with multiple, changing goals in a dynamic, unpredictable environment.

1. Why this paper?

Since 1985, a new wave has emerged in the study of Artificial Intelligence (AI). At the same moment at which the popular, general belief is that AI has been a "failure", many insiders believe that something exciting is happening, that new life is being brought to the field. The new wave has been termed "behavior-based AI" as opposed to mainstream "knowledge-based AI", or also "bottom-up AI" versus "top-down AI". Finally, the term "animat approach", which was coined by Wilson (1985), is also frequently used. This paper attempts to describe what Behavior-Based AI is about and how it differs from Knowledge-Based AI. It argues that Behavior-Based AI poses problems in a different way, investigates interesting new techniques and applies a set of different criteria for success.

Several people have tried to define Behavior-Based AI, among others Brooks (1991), Wilson (1991) and Meyer (1991). There are several reasons for giving it yet another try. First of all many researchers are still sceptical about the approach. Some claim that it isn't very different from what they have been doing all along. Others are still not convinced that the approach is founded and scientific. A second reason is that this account is different from the papers listed above. Brooks, being one of the main originators of this new approach, presents a picture which is restricted towards robotic forms of intelligence (Brooks, 1991).

This paper presents a more general perspective. It argues that the behavior-based approach is appropriate for the class of problems that require a system to autonomously fulfill several goals in a dynamic, unpredictable environment. This includes applications such as interface agents (Maes, 1992b), process scheduling (Malone et.al., 1988), and so on. Wilson's account (1991) focuses on a scientific methodology for Animat research, while Meyer (1991) aims to give an overview of the research performed so far.

Defining a new approach is a difficult and tricky thing to do. In particular, it is dangerous to draw solid lines between the different approaches and to force examples of research to fit into one of them. For the sake of clarity the remainder of the paper presents Knowledge-Based AI and Behavior-Based as two extremes. The reader should keep in mind that concrete examples of research are often situated along the continuum between the two extreme positions presented here and that those intermediate positions are valid ones to adopt.

The paper is structured as follows. Section 2 distinguishes the goal and emphasis of Knowledge-Based AI with those of Behavior-Based AI. Section 3 discusses the solutions investigated by the Knowledge-Based Approach. The next chapter lists some of the key insights which underlie the Behavior-Based approach. Section 5 elaborates upon the solutions adopted by Behavior-Based AI. Section 6 discusses some examples contrasting the two approaches. Finally section 7 contains some critical comments about the progress made so far.

2. Goals of the Two Approaches to AI

The goal of both Knowledge-Based AI and Behavior-Based AI is to synthesize computational forms of intelligent systems. Both approaches attempt to model intelligent phenomena such as goal-directed

[0]Reprinted with permission from: "From Animals to Animats 2: Proceedings of the Second International Conference on Simulation of Adaptive Behavior" edited by J.A. Meyer, H. Roitblat and S. Wilson, MIT Press 1993.

behavior, prediction, learning, communication and cooperation. Knowledge-Based AI has traditionally emphasized the modeling and building of systems that "know" about some problem domain. These systems model the domain and can answer questions about this problem domain, often involving extensive problem solving and reasoning. Behavior-based AI on the other hand has emphasized the modeling and building of systems that "behave" in some problem domain.

More specifically, "main-stream" Knowledge-Based AI can be characterized as typically studying systems with the following characteristics:

1. They model isolated, often advanced, competences (e.g. medical diagnosis, chess playing, etc). They rather provide "depth" than "width" in their expertise.

2. They are "closed", in the sense that there is no direct interaction with the problem domain about which they encode knowledge and solve problems. Their only connection with the environment is the user. The user recognizes a problem in the domain, describes it to the system in the symbolic language which the system understands. The system then returns a symbolic description of an answer or solution, which then has to be implemented by the user in the actual domain.

3. They deal with one problem at a time. Often, they do not have time constraints for solving the problem (although the user might) and do not have to deal with interrupts. From the system's point of view the problem domain does not change while the system is computing. They also do not have to deal with multiple problems simultaneously. They are given one problem at a time by the user.

4. They have declarative "knowledge structures", which model aspects of the domain of expertise. All of the internal structures, apart from an interpreter are static. The system is only active when a problem is posed by the user, in which case the interpreter uses the static knowledge structures to determine the solution to the problem.

5. They are not usually concerned with the developmental aspect, or how the knowledge structures got there in the first place and how they should change over time. They do not have to be adaptive to changing situations (components breaking down, etc). At the most some form of knowledge compilation or knowledge optimization is incorporated.

6. In the few cases where an autonomous system (e.g. a robot or an interface agent) is being modelled, a central system which has all of the above characteristics is augmented with a perception module and an execution module which take over part of the role of the human interface. The perception module has to recognize the current situation and translate it into a symbolic description for the central system. The "problem" or goal is usually still specified in symbolic terms by a human. The execution module is responsible for "implementing" the description of the solution produced by the central system in the problem domain.

In contrast, Behavior-Based AI has typically studied the following type of system:

1. It has multiple integrated competences. Typically the competences are lower-level (as opposed to expert level). For a robot these are competences such as locomotion, navigation, survival, collecting objects, etc. For other systems these might be other simple competences, like reacting in a market system by simple bidding and buying behaviors (Malone et.al., 1988) or executing a simple routine in the case of an interface agent (Maes, 1992b) (Kozierok and Maes, 1992).

2. The system is "open" or "situated" in its environment. It is directly connected to its problem domain through sensors and effectors. It can affect or change this domain through its output. The problem domain is typically very dynamic, which means that the system has a limited amount of time to act. The domain is usually also very complex. Unpredictable events happen all the time. It typically also involves other acting agents (human and/or artificial).

3. The emphasis is on autonomy: the system is completely self-contained. It has to monitor the domain and figure out by itself what the problem to be solved next is. Typically it has to deal with many conflicting goals simultaneously.

4. Rather than on knowledge, the emphasis is on the resulting behavior of the system. Its internal structures are active "behavior producing" modules as opposed to static "knowledge structures". They do not have to be initiated by a goal formulation from the user. It is less important that the system can answer questions about its problem domain (such as how to solve particular problems). It is also less important that the user is able to inspect the internal structures and identify those that are responsible for particular aspects of the behavior. For example, it is acceptable for goals or planning to be emergent observable properties, which cannot be attributed to particular internal structures.

5. Finally, there is a strong emphasis on "adaptation" and on a "developmental approach". This often means that the system improves its own structures (and thus behavior) over time based on its own experience in the environment. In other

cases, this means that an incremental approach is taken: the user gradually evolves a more sophisticated system by adding structure to an already existing "working" system.

One can conclude from the above discussion that it is hard to compare the Knowledge-Based and Behavior-Based approach because they typically study different classes of problems: knowledge versus behavior, a single high-level competence versus a range of low-level competences, user-driven computation versus autonomous systems, and so on. Both classes of problems are interesting in their own right. So far, neither approach has shown much success dealing with the problem classes concentrated on by the other approach. As long as it remains to be seen whether either one will be able to broaden its domain of success in significant ways, both types of research are necessary and complementary.

3. Solutions investigated by Knowledge-Based AI

The difference between Knowledge-Based AI and Behavior-Based AI lies not only in the problems that are studied, but also in the techniques and solutions that are explored. The solutions typically adopted in main-stream AI projects can be characterized as follows:

Modular Decomposition

The intelligent system is decomposed along "functional modules" such as perception, execution, natural language communication (the peripheral components), a learner, planner and inference engine (the central systems components). These modules are typically developed independently. They rely on the "central representation" as their means of interface. The central representation includes things such as beliefs (updated by perception, also read and augmented by the inference engine and the natural language component), desires (or goals) and intentions (produced by planner).

Approach

Typically all of these functional components are as general and domain-independent as possible. The hope is that the same functional components can be used for different problem domains (a general domain-independent planner, learner, etc). The only component which needs to be adapted is the central representation, which contains domain-specific information such as heuristic knowledge.

Role of Representation

The key issue on which emphasis is laid is a complete, correct internal model, a perfect copy of the world (with all its objects and relationships) inside the system, which the system can rely on to predict how the problem can be solved.

Organization

The organization of the different modules within the system is completely sequential. The modules take turns being "active" or processing and changing the internal representations. Perception and inference first update the internal model (beliefs and goals). After that, planning or problem solving produce a description of the solution to the problem (a plan or the answer to a question). Finally either the execution module or a human implements the solution in the domain (the latter one having more knowledge and understanding of the situation than the former one).

Model of Activity

Activity is modeled purely as the result of a "deliberative thinking" process. The central system evaluates the current situation (as represented in the internal model) and uses a search process to systematically explore the different ways in which this situation can be affected so as to achieve the desired goals.

Role of Learning

Learning typically consists of compilation or reformulation of what the system already knows. For example, the system might cache a plan for later reuse. Very seldom is there learning of new information or correction of existing knowledge based on environmental feedback. This implies that the programmer is completely responsible for creating an initial complete and correct model.

The Knowledge-Based approach has produced several successes, in particular in the area of knowledge systems assisting experts with the modelling of and reasoning about a specific problem domain. For a range of reasons the approach has proven nonsatisfactory when dealing with the class of problems Behavior-Based AI is interested in. Several experiments which attempted to use the above approach for constructing autonomous systems operating in dynamic environments have run into the following problems[1].

- The resulting systems can be slow, because of the sequential processing of information and because of the high computational cost involved in maintaining a model and doing general perception and planning.
- The resulting systems can be inflexible. They have difficulty reacting fast to changes in the environment. They do not deal very well with unexpected

[1] See for example the literature on the Shakey project (Nilsson, 1984) or the more recent Ambler project (Bares et.al., 1989) for examples of this approach in the study of autonomous robots. Sullivan and Tyler (1991) present an example of this approach in the area of interface agents.

opportunities or contingencies. This is partly due to the fact that changes have to propagate through many sequential layers before they affect the actions taken by the system. Another reason is that these systems are built on the assumptions that few or no unpredictable changes will happen.

- The resulting systems tend to be brittle. They fail in situations that only differ slightly from the ones they are programmed for. They do not show graceful degradation of performance as components break down. For example, if the perception module cannot make sense of the current situation, the whole system might break down.

- In practice it proves to be hard to relate the symbols in the internal model (representing objects and relations in the domain) to physical stimuli. It is hard to keep track of object identities, often the sensor data are ambiguous, erroneous, inconsistent, and so on.

- In practice it also proves to be hard to hand-build a complete and consistent model of the environment which the system can rely on to make predictions.

- Another practical problem is that of combinatorial explosions. General planning and problem solving have proven to be a computationally expensive process (Chapman, 1987).

- Several theoretical problems have come up, which remain unsolved in satisfactory ways. Examples of such problems are the frame problem and the problem of non-monotonic reasoning.

4. Important Insights of Behavior-Based AI

The methods developed by Behavior-Based AI in response to the problems of Knowledge-Based AI techniques listed above, are grounded in two important insights[2]:

- Looking at complete systems changes the problems often in a favorable way.

- Interaction dynamics can lead to emergent complexity.

A first realization is that viewing the problem of building an intelligent system in its context can make things a lot easier. This observation is true at several levels:

1. The intelligent functions which are being modelled are part of a complete intelligent system. Building systems in an integrated way (rather than developing modules implementing these functions independently) often makes the task a lot easier. For example, a system which can learn has to rely less on planning (because it can cache computed plans). A system which has sensors and actuators can perform tests in the environment and as such has less of a need for modelling and inference. A system which has sensors has an easier job disambiguating natural language utterances and so on.

2. The complete system is part of some environment, it is situated in some space. This implies that there is less of a need for modeling, because the "world is its own best model" (Brooks, 1991). The environment can also be used as an external memory (e.g. for reminding the system which tasks still have to be performed and which ones it already did perform (Suchman, 1987). The environment usually has particular characteristics which can be exploited by the system (offices consist of vertical walls and horizontal floors, doors typically have a particular size, etc). These "habitat constraints" can be exploited by the system, making its task much easier (Horswill, 1992).

3. The system is not only situated in space, but also in time. This implies that the system can evolve itself so as to become better at its task, if time and the particular task admits (either through individual learning or some sort of artificial evolution). Time also allows for the construction of an iterative, incremental solution to a problem. For example, a natural language system situated in time does not need to disambiguate every utterance. It can go back and forth asking questions or making particular remarks which will help to gradually disambiguate whatever the other speaker wants to convey.

4. Finally the system is typically also part of a society. Other agents in the same environment are dealing with similar problems. Therefore there is no need for the agent to figure everything out by itself. For example, a mobile robot could use the strategy of closely following a person passing by, so as to achieve the competence of navigating in an office environment without bumping into things. Kozierok and Maes (1992b) report on some experiments in which interface agents learned to perform certain tasks by observing and imitating users.

As a consequence of the above ideas, Behavior-Based AI has concentrated on modelling systems in their context. While traditional AI has concentrated on simulated toy examples, Behavior-Based AI has built "real" systems which solve an actual (small) problem in a concrete environment.

A second major insight of Behavior-Based AI is that interaction dynamics among simple components can lead to emergent complexity (see also Resnick, 1992). Also this idea applies at several different levels (Brooks, 1991b):

1. Simple interaction dynamics between the system and its environment can lead to emergent struc-

[2]Notice that I do not credit Behavior-Based AI with the discovery of these insights.

ture or emergent functionality. Simon (1968) gives an example of an ant on the beach. He notes that the complexity of its behavior is more a reflection of complexity of environment than of its own internal complexity and postulates that the same might be true of human behavior. Agre (1991) shows how behavior as complex as goal-directed action sequences can be an observable, emergent property of the interaction dynamics between the environment and a reflex-guided person. This means that often it is sufficient to study the particular properties of the environment and find an interaction loop, a set of simple feedback or reflex mechanisms, which will produce the desired behavior. One of the implications is that we need a better understanding of environments (Horswill, 1992), (Wilson, 1991). It also means that if we want to prove aspects about the resulting performance of Behavior-Based systems, we have to model these systems as well as their environments.

2. Simple interaction dynamics between the components of the system can lead to emergent structure or emergent functionality. For example, Mataric's wall-following robot does not have a single component to which the expertise of wall-following can be attributed (Mataric, 1991). One module is responsible for steering the robot towards the wall when the distance to the wall is above some threshold while another module is responsible for steering the robot away from the wall when the distance is below some threshold. Neither one of these modules is primarily "responsible" for the wall following behavior. It is their interaction dynamics which makes the robot follow walls reliably. In Maes' networks (1990), none of the component modules is responsible for action selection. The action selection behavior is an emergent property of some activation/inhibition dynamics among the primitive components of the system.

3. Finally interaction dynamics between the component systems in a social system can lead to emergent structure or functionality. Deneubourg (1991, 1992) describes how social insects following simple local rules can produce emergent complexity such as a path to a food source, food foraging trees, etc. Malone's collection of autonomous bidding systems solves the complicated task of process-processor allocation (Malone et.al., 1988). Finally, anthropologists have studied the social construction of different concepts and methods (Suchman, 1987) (Shrager and Callanan, 1991).

Important is that such emergent complexity is often more robust, flexible and fault-tolerant than programmed, top-down organized complexity. This is the case because none of the components is really in charge of producing this complexity. None of the components is more critical than another one. When one of them breaks down, the system demonstrates a graceful degradation of performance. Since all of the components interact in parallel, the system is also able to adapt quicker to environmental changes. Often the system explores multiple solutions in parallel, so that as soon as certain variables change, the system is able to switch to an alternative way of doing things. For example, in Maes' system (1990) several sequences of actions are evaluated in parallel, the best one determining the behavior of the agent. Also in Malone's system (Malone et.al., 1988) several mappings of processes to machines can be said to be explored in parallel.

5. Solutions Investigated by Behavior-Based AI

In section 3 we listed the techniques adopted by Knowledge-Based AI and discussed why they proved to be inadequate for building autonomous systems situated in dynamic environments. This section contrasts these techniques with those adopted by Behavior-Based AI.

Modular Decomposition

Instead of building general functional modules like perception and planning, Behavior-Based AI develops competence modules, modules which are an expert at (and are responsible for) a particular small task-oriented competence. These modules interface to one another via extremely simple messages (rather than a common representation of beliefs, etc). The communication is almost never of a "broadcast" nature, but happens rather on a one-to-one basis. Typically the messages consist of activation energy, or simple suppression and inhibition signals, or simple tokens in a restricted language. Each of the modules is directly connected to relevant sensors and actuators.

Approach

There are no "general" or task-independent modules. There is no general perception module, no general planner, etc. Each of the competence modules is responsible for doing all the representation, computation, "reasoning", execution, etc, related to its particular competence. For example, an obstacle avoidance module might need one bit of information to represent whether an obstacle is perceived or not, and it might do some very simple computation to decide how an obstacle should be avoided. Competence modules are self-contained, black boxes. They might employ completely different techniques (even different hardware) to achieve their competence. Part of the reason for this more pragmatic approach is a pessimistic vision about whether it is possible at all to build a general vision system, a general planner, etc (a view also expressed in (Minsky, 1986).

Role of Representation

There is much less emphasis on modeling the domain. First of all, there is no central representation shared by several modules. The system also does not attempt to integrate the information from different sensors into one coherent, objective interpretation of the current situation. Instead every task-oriented module represents whatever it needs to represent to achieve its competence. These representations are not related and might be inconsistent or redundant. Within one competence module, the usage of representations is minimized in favor of employing the environment as a source of information (and a determiner of action). The representations within one module are often of a less propositional, objective and declarative nature than those employed in Knowledge-Based AI. For example they might index objects according to the features and properties that make them significant to the task at hand (Agre, 1991) rather than their identities. They can be of a numeric, procedural or analog nature. Often a lot of task-specific "problem solving" is performed in the perception part of a particular competence (Steels, 1990b) (Chapman, 1992) (Ballard, 1989).

Organization

The systems built are highly distributed. All of the competence modules operate in parallel. None of the modules is "in control". However, some simple arbitration method is often included in order to select or fuse multiple conflicting actuator commands. This arbitration network might be a winner-take-all network, as in (Maes, 1990) or a hardcoded priority scheme as in (Brooks, 1986). Because of its distributed operation, a behavior-based system is typically able to react very fast to changes in the environment or changes in the needs of the system.

Model of Activity

Activity is not modelled as the result of a deliberative process. Instead complex and goal-directed activity is modeled as an emergent property of the interaction among competence modules internally, and among competence modules and the environment. There is no internal structure corresponding to "the plan" of the system.

Role of Learning

Learning and development are considered crucial aspects of a Behavior-Based System (Wilson, 1985) (Maes, 1992). Building an adaptive system that will develop into one that achieves the tasks, is often considered a better approach than building a static system which will not change when the environment or task changes (e.g. a robot breaking one of its legs). In some systems, the evolution (towards increasingly more sophisticated behavior) is simulated by the programmer, e.g. by incrementally adding more structure to existing successful systems (Brooks, 1991). Other systems employ artificial evolution (Koza, 1991) or learning by the individual (Maes, 1992) (Maes and Brooks, 1990) (Wilson, 1985) (Drescher, 1991) (Kaelbling, 1992) (Sutton, 1991). In almost all cases, the system concentrates on learning new information (or behavior) from its environmental, rather than compiling existing information. The learning algorithms are implemented in a distributed way: typically a similar learning algorithm runs in different competence modules. Related to the idea of learning is that of redundancy: often the system has multiple modules for a particular competence. Experience sorts out which of these modules implements the competence in a more reliable way (Maes, 1992) (Payton et.al., 1991) (Drescher, 1991).

Systems built using the above principles suffer less from the problems listed in section three. They act fast, because (1) they have less layers of information processing, (2) they are more distributed and often non-synchronized, and (3) they require less expensive computation. They are able to deal with unforeseen situations (opportunities as well as contingencies), because they rely much more on the environment as a source of information and a determiner of action. They are less brittle, because (1) none of the modules is more critical than the others, (2) they do not attempt to fully understand the current situation, (3) they incorporate redundant methods and (4) they adapt over time. They have less trouble relating representations to sensor stimuli, because they do not attempt to maintain objective representations of the environment. Finally they are not prone to problems of combinatorial explosions, because they do not employ traditional search processes.

6. Three Examples Contrasting the Two Approaches

A Mobile Robot

Consider a mobile surveillance robot which has to watch over some offices. Its task requires that it navigates from room to room. The Knowledge-Based version of this robot could work as follows. The perception module processes the different sensor data and integrates them into a representation of the environment. It attempts to update this model as often as possible. The model includes information such as the location of the robot in the environment, the location and type (often even identity) of other objects in this environment such as chairs, tables, etc. The model is used by the planning module to decide how to fulfill the goal of finding the door in the current room, while avoiding obstacles. The planner goes through a systematic search to produce a list of actions which will according to the model fulfill both goals. The execution module executes this plan while possible checking at certain points whether things are going as predicted. If not, control is returned to the planner.

A Behavior-Based robot for the same task could be constructed in the following way. In an incremental way, several modules would be implemented corresponding to the different competences necessary for the task: a module for recognizing and going through doors, a module for wall following (actually wall following is often modeled as an emergent property of two to three lower-level modules), a module for obstacle avoidance (or even a couple redundant ones, using different sensors, since this is a very critical competence), and so on. All of these modules operate in parallel. A simple arbitration scheme (some simple suppression and inhibition wires among these modules) suffices to implement the desired priority scheme: the obstacle avoidance modules always have priority over going through doors which has priority over wall following. This robot does not plan a course of action. However from an observer's point of view it will appear to operate in a systematic, rational way. Brooks (1986)(1991) has argued convincingly in writing and in actual demonstrations, which of the two above robots will be more able to fulfill the task in a robust and reliable way.

An Interface Agent

Consider the problem of building a "software agent" or "interface agent" which assists the user with certain computer-based tasks. Its goal is to offer assistance to the user and automate as many of the actions of the user as possible. Knowledge-Based AI approaches this problem in the following way (Sullivan and Tyler, 1991). The agent is given an elaborate amount of knowledge about the problem domain by some Knowledge Engineer. This knowledge contains: a model of the user and possibly the user's organization, a model of the tasks the user engages in, including a hierarchical specification of the subtasks, knowledge about the vocabulary of these tasks, and so on. At run time, the agent uses this knowledge to recognize the intentions and plans of the user. For example, if a UNIX user enters a command like "emacs paper.tex", the system deduces that the user is planning to produce a written document. It then plans it's own course of action (the goal being to assist the user), which for example might consist of the action sequence: the text formatting command "latex paper.tex", followed by the preview command "xdvi papr.dvi" and the printing command "lpr paper.dvi". The problems with this approach are exactly the same ones as those for mobile robots (cfr. list in section three): it is hard to provide such a complete and consistent model, the model is quickly outdated (as the user's ways of performing tasks change). Because of the computational complexity of the approach, the system would react very slow. All sorts of unpredicted events might take place which the agent cannot deal with (the user might change his/her mind about what to do in the middle of things, or might perform tasks in unorthodox non-rational ways), etc.

Instead a Behavior-Based interface agent can be built as follows (Maes, 1992) (Kozierok and Maes, 1992). Several competence modules are constructed which are experts (or try to become experts) about a small aspect of the task. For example, one module might be responsible for invoking a particular command (like "lpr") at a particular moment. The agent is situated in an environment containing an ideal source for learning: the user's behavior. Each of the modules gathers information by observing the user and keeping statistics about a particular aspect of the user's behavior. For example, the above mentioned module will keep track of the situations in which the user executed the "lpr" command. Whenever a new situation comes up which is very similar to one of one or more memorized situations, it actually offers to the user to execute the "lpr" command. If we have several experts for the different commands listed above, each of these will know when to become active and offer their assistance to the user. From an observer's point of view, it will seem as if the system "understands" the intentions of the user, as if it knows what the task of producing a document involves. Nevertheless, the action sequences are just an emergent property of a distributed system. The system will smoothly adapt to the changing habits of the user, will react in a fast way, will never completely break down, and so on.

A Scheduling System

Finally, consider the problem of building a scheduling system, whose goal it is to allocate processes to processors in real-time. Again the domain is a very dynamic one: new processing jobs are formulated in different machines all the time. The decision to be made is whether to run these processes locally or on a different machine, the global goal being to minimize the average amount of time it takes to run a process. The loads of the different available machines vary continuously. Certain machines might suddenly become unavailable for scheduling processes, requiring a rescheduling of the jobs that were running on those machines at the time, and so on. A Knowledge-Based system for this task would contain a lot of knowledge about scheduling and about the particular configuration of machines and typical processing jobs at hand. The system would update its representation of the current situation as often as possible. This requires gathering all the data from the different machines in the network on whether they are still available, what their workload is, which processes they are running, which new processes were formulated on them, etc. Once all this information has been centralized, the system would perform a systematic search (possibly involving some heuristics) for the most optimal allocation of processes to processors. Once that schedule has been produced, the processing jobs can actually be sent to the different machines that they have been assigned to. This centralized way of solving the problem is present in the majority of the traditional work in this area (Kleinrock and Nilsson, 1981).

Malone has proposed a different solution to this problem (Malone et.al, 1988), which one could call Behavior-Based. In his "Enterprise" system, each of the machines in the network is autonomous and in charge of its own work load. The system is based on the metaphor of a market. A machine on which a new processing task originates, sends out a "requests for bids" for the task to be done. Other machines respond (if they feel like it) with bids giving estimated completion times which reflect their speed and currently loaded files. For example, if the task to be performed is a graphics rendering job and some machine has that software loaded, it will be more interested in running the new job (because it does not have do waste time and space loading the necessary software). The machine which sent out the request for bids will collect the bids it receives over some small period of time and allocate the job to the machine which made the best bid (either remote or local). This distributed scheduling method was found to have several advantages. The system is very robust because none of the machines is more critical than another one (there is no central scheduler). A user can make a machine unavailable for external processing jobs at run-time. The whole system will adapt smoothly to this unexpected situation. The system is very simple and yet very flexible in terms of the kind of factors it can take into account.

7. Discussion

Behavior-Based AI represents an exciting new approach to the study of intelligence. So far, Behavior-Based AI has demonstrated several "proofs-of-concept" of its approach. In particular, successes have been booked in the area of autonomous, situated systems. Several prototypes have been built which have shown to solve some reasonable difficult task in a real, dynamic domain. These initial results are very promising, but are far from representing a solid, systematic methodology.

In order for the new approach to be more founded, more fundamental research has to be undertaken. First of all, we need to understand the classes of problems Behavior-Based AI is trying to deal with much better, so that it becomes possible to critically compare particular systems and proposals. For example, many different models of action selection have been proposed, but unless we understand the problem of action selection better, and have a list of desiderata for solutions, we do not have any ground to compare the different proposals[3].

Aside from better evaluation criteria, we need a better understanding of the underlying principles of Behavior-Based AI. Without an underlying theory, it will not be possible to scale the approach. In particular, it is important to understand the mechanisms and limitations of emergent behavior. How can a globally desired structure or functionality be designed on the basis of local rules? What are the conditions and limitations under which the emergent structure is stable? and so on. Some first steps towards a theory of emergent functionality have been proposed, using tools from complex dynamics (Steels, 1991) (Kiss, 1991).

[3]Some people have started making such lists, e.g. (Brooks, 1991b), (Tyrell, 1992) and (Maes, 1990, 1990).

Acknowledgements

Parts of this paper were prepared for a public discussion on the topic of Behavior-Based AI with Marvin Minsky and Mike Travers. Karl Sims provided comments on an earlier draft.

References

Agre P.E, 1991, The Dynamic structure of Everyday Life, Cambridge University Press.

Ballard D.H, 1989, Reference Frames for Animate Vision, Proceedings of IJCAI-89 conference, Detroit.

Bares J., Hebert M., Kanade T., Krotkov E., Mitchell T., Simmons R., Whittaker W., 1989, Ambler: An Autonomous Rover for Planetary Exploration, IEEE Computer, June 1989.

Brooks R.A., 1986, A Robust Layered Control System for a Mobile Robot, IEEE Journal of Robotics and Automation, RA-2, April.

Brooks R.A., Maes P., Mataric M.J., and More G., 1990, Lunar Base Construction Robots, Proceedings of the IROS-90 conference, Japan.

Brooks R.A., 1991, Intelligence without Reason, Computers and Thought lecture, Proceedings of IJCAI-91, Sidney, Australia.

Brooks R.A., 1991b, Challenges for Complete Creature Architectures, In: From Animals to Animats, Proceedings of the First International Conference on the Simulation of Adaptive Behavior, Edited by Meyer J.-A. and Wilson S.W., MIT-Press, 1991.

Chapman D., 1987, Planning for Conjunctive Goals, Journal of Artificial Intelligence, 32, 3.

Chapman D., 1992, Vision, Instruction and Action, MIT-Press, in press.

Deneubourg J.L., Goss S., Franks N., Sendova-Franks A., Detrain C., Chretien L., 1991, The Dynamics of Collective Sorting: Robot-like Ants and Ant-like Robots, In: From Animals to Animats, Proceedings of the First International Conference on the Simulation of Adaptive Behavior, Edited by Meyer J.-A. and Wilson S.W., MIT-Press, 1991.

Deneubourg J.L, Theraulaz G. and Beckers R., 1992, Swarm-Made Architectures, In: Toward a Practice of Autonomous Systems, Proceedings of the First European Conference on Artificial Life, edited by F.J. Varela and P. Bourgine, MIT-Press.

Drescher G.L, 1991, Made-Up Minds: A Constructivist Approach to Artificial Intelligence, MIT-Press.

Holland J.H., 1986, Escaping Brittleness: the Possibilities of General-Purpose Learning Algorithms applied to Parallel Rule-Based Systems, In: Machine Learning, an Artificial Intelligence Approach, Volume II, edited by R.S. Michalski, J.G. Carbonell and T.M. Mitchell, Morgan Kaufmann.

Horswill I., 1992, Characterizing Adaptation by Constraint, In: Toward a Practice of Autonomous Systems, Proceedings of the First European Conference on Artificial Life, edited by F.J. Varela and P. Bourgine, MIT-Press.

Kaelbling L.P., 1992, An Adaptable Mobile Robot, In: Toward a Practice of Autonomous Systems, Proceedings of the First European Conference on Artificial Life, edited by F.J. Varela and P. Bourgine, MIT-Press.

Kiss G., 1991, Autonomous Agents, AI and Chaos Theory, In: From Animals to Animats, Proceedings of the First International Conference on the Simulation of Adaptive Behavior, Edited by Meyer J.-A. and Wilson S.W., MIT-Press, 1991.

Kleinrock L. and Nilsson A., 1981, On Optimal Scheduling Algorithms for Time-Shared Systems, Journal of the ACM, 28, 3, July 1981.

Koza J.R., 1991, Evolution and Co-evolution of Computer Programs to Control Independently-Acting Agents, In: From Animals to Animats, Proceedings of the First International Conference on the Simulation of Adaptive Behavior, Edited by Meyer J.-A. and Wilson S.W., MIT-Press, 1991.

Kozierok R. and Maes P., 1992, A Learning Interface Agent for Scheduling Meetings, Proceedings of the ACM SIGCHI International Workshop on Intelligent User Interfaces, ACM, Orlando, Florida.

Maes P., 1990, Situated Agents Can Have Goals, In: Designing Autonomous Agents: Theory and Practice from Biology to Engineering and Back, edited by P. Maes, MIT Press.

Maes, P., 1990b, Designing Autonomous Agents: Theory and Practice from Biology to Engineering and Back, MIT-Press

Maes P. and Brooks R.A., 1990, Learning to Coordinate Behaviors, Proceedings of AAAI-90, Boston.

Maes P., 1992, Learning Behavior Networks from Experience, In: Toward a Practice of Autonomous Systems, Proceedings of the First European Conference on Artificial Life, edited by F.J. Varela and P. Bourgine, MIT-Press.

Maes P., 1992b, Learning Interface Agents, Submitted to the ACM SIGCHI International Workshop on Intelligent User Interfaces, ACM, Orlando, Florida.

Malone T.W., Fikes R.E., Grant K.R., and Howard M.T., 1988, Enterprise: A Market-like Task Scheduler for Distributed Computing Environments, In: The Ecology of Computation, edited by B. Huberman, North-Holland.

Mataric M., 1991, Behavioral Synergy without Explicit Integration, In: Special Issue of SIGART on Integrated Cognitive Architectures, Volume 2, Number 4.

Meyer J.-A. and Guillot A., 1991, Simulation of Adaptive Behavior in Animats: Review and Prospects, In: From Animals to Animats, Proceedings of the First International Conference on the Simulation of Adaptive Behavior, Edited by Meyer J.-A. and Wilson S.-W., MIT-Press, 1991.

Minsky M., 1986, the Society of Mind, Simon and Schuster, New York.

Nilsson N., 1984, Shakey the Robot, SRI A.I. Center Technical Note 323.

Payton D.W., Keirsey D., Krozel J. and Rosenblatt K., 1991, Do Whatever Works: A Robust Approach to Fault-Tolerant Autonomous Control, to appear in Journal of Applied Intelligence, Vol. 3, 1992.

Resnick M., 1992, Beyond the Centralized Mind Set: Explorations in Massively Parallel Microworlds, Ph.D. Thesis, MIT Media-Laboratory Epistemology and Learning Group.

Roitblat H.L., 1991, Cognitive Action Theory as a Control Architecture, In: From Animals to Animats, Proceedings of the First International Conference on the Simulation of Adaptive Behavior, Edited by Meyer J.-A. and Wilson S.W., MIT-Press, 1991.

Shrager J. and Callanan M., 1991, Active Language in the Collaborative Development of Cooking Skill, Proceedings of the Cognitive Science Conference, Lawrence Erlbaum.

Simon H., 1969, The Sciences of the Artificial, MIT-Press.

Steels L., 1990, Cooperation between Distributed Agents through Self-Organization, In: Decentralized AI, edited by: Y. Demazeau and J.-P. Muller, Elsevier, North-Holland.

Steels L., 1990b, Exploiting Analogical Representations, In: Designing Autonomous Agents: Theory and Practice from Biology to Engineering and Back, edited by P. Maes, MIT Press.

Steels L., 1991, Towards a Theory of Emergent Functionality, In: From Animals to Animats, Proceedings of the First International Conference on the Simulation of Adaptive Behavior, Edited by Meyer J.-A. and Wilson S.W., MIT-Press, 1991.

Suchman L.A., 1987, Plans and Situated Actions: The Problem of Human-Machine Communication, Cambridge University Press.

Sullivan J.W., and Tyler S.W. (editors), 1991, Intelligent User Interfaces, ACM Press.

Sutton Richard S., 1991, Reinforcement Learning Architectures for Animats, In: From Animals to Animats, Proceedings of the First International Conference on the Simulation of Adaptive Behavior, Edited by Meyer J.-A. and Wilson S.W., MIT-Press, 1991.

Tyrrell T., 1992, Defining the Action Selection Problem, In: Proceedings of the 14th Conference of the Cognitive Science Society.

Wilson S.W., 1985, Knowledge Growth in an Artificial Animal, In: Proceedings of the First International Conference on Genetic Algorithms and their Applications, edited by Greffenstette, Lawrence Erlbaum Assoc.

Wilson S.W., 1991, The Animat Path to AI, In: From Animals to Animats, Proceedings of the First International Conference on the Simulation of Adaptive Behavior, Edited by Meyer J.-A. and Wilson S.W., MIT-Press, 1991.